Article

Not peer-reviewed version

An Energy-Efficient Logistic Drone Routing Method considering Dynamic Drone Speed and Payload

Kunpeng Wu , Shaofeng Lu * , Haoqin Chen , Minling Feng , Zenghao Lu

Posted Date: 18 March 2024

doi: 10.20944/preprints202403.1028.v1

Keywords: Drone routing problem; Logistics drone; Mixed-integer linear programming; Adaptive large neighborhood search

 doi:10.20944/preprints202403.1028.v1

Article

An Energy-Efficient Logistic Drone Routing Method Considering Dynamic Drone Speed and Payload

Kunpeng Wu [1], Shaofeng Lu [1,*], Haoqin Chen [1], Minling Feng [1] and Zenghao Lu [2]

[1] South China University of Technology; kunpeng.wu@outlook.com
[2] Fujian Zhongli Technology Co.; zenghao.lu@qq.com
* Correspondence: lushaofeng@scut.edu.cn; Tel.:+8613626191230

Abstract: Unmanned aerial vehicle (UAV), or drone is recognized for its potential to improve efficiency and address last-mile delivery issues. As a result, there has been a lot of activity in recent years in the field of drone scheduling and routing. Unlike the vehicle routing problem, drone route design is difficult due to several operational characteristics, such as speed optimization, multitrip operation, and energy consumption estimation. On the one hand, drone energy consumption is a complex nonlinear function of both speed and payload in practice. On the other hand, the high operating speed of drones can significantly curtail the drone range, thereby limiting the efficiency of drone delivery systems. Most of the existing drone delivery models either assume constant drone speed or do not consider the effect of drone speed and parcel weight on energy consumption, leading to costly or energy-infeasible routes. This paper addresses the trade-off between speed and flight range in a multi-trip drone routing problem with variable flight speeds (DRP-VFS), in which a team of homogeneous drones is employed for delivery services. We propose a new model to particularly consider energy constraints using a nonlinear energy consumption model and treat drone speeds as decision variables so that various drone speeds can be adopted in applications. The DRP-VFS is initially formulated as mixed-integer linear programming (MILP) to minimize total energy consumption. To solve large-scale instances, we propose a three-phase adaptive large neighborhood search (ALNS) algorithm. The experimental results demonstrate that suboptimal solutions can be found effectively in practical scenarios using the proposed method. Furthermore, results indicate that operating drones at variable speeds leads to about 21% of energy savings compared to fixed speeds, boasting advantages in cost-savings and range extensions.

Keywords: drone routing problem; logistics drone; mixed-integer linear programming; adaptive large neighborhood search

1. Introduction

Over the past few decades, the development of technology, especially carbon fiber and lithium polymer batteries, has facilitated the use of drones for last mile delivery. For example, Zipline has completed over 14,000 medical supply deliveries in Rwanda since 2016 ([1]). In 2019, Google's Wing and Amazon received Federal Aviation Administration (FAA) approval to begin commercial delivery by drone ([2,3]).

In recent years, researchers have shown a growing interest in last-mile delivery that incorporates drones ([4,5]). The drone delivery problems can be categorized as drone-only or truck-drone tandem problems. The first uses only drones for parcel delivery, while the second uses both drones and trucks to deliver. The alternative objectives of drone delivery problems are to minimize delivery time or minimize operational cost ([6]). Numerous variants of these problems have been suggested, such as heterogeneous drones ([7]) or multiple trucks ([8,9]).

Compared with traditional truck transportation, drone delivery is faster and less affected by road systems, in addition, it can save a lot of labor and time costs ([10]). These benefits allow logistics companies to dispatch drones for parcel delivery. On the other hand, a drone's endurance is constrained by its battery's capacity, which is further impacted by parcel weight, speed, and weather conditions ([11–13]). In the literature involving drone delivery, one typical assumption is that drones fly at constant or maximum speed, regardless of their payload or speed. Thus, speed is not taken into

 doi:10.20944/preprints202403.1028.v1

consideration during the decision-making procedure. In fact, operating drones to fly at a fixed speed may lead to increased energy consumption, thus resulting in an energy-infeasible route or an inability to serve long-distance customers because the energy consumption of drones is severely impacted by parcel weight and speed. Therefore, it is crucial to consider the impact of payload and speed on endurance in drone routing problems.

The motivation for considering flight speeds as decision variables is the influence of drone energy consumption on flight endurance and range. Figure 1(a) shows the power rate of drones under different payloads, using the model suggested by [12]. The circular marks indicate optimal drone speeds that maximize the drone endurance under any onboard battery capacity, which can be seen more clearly from Figure 1(c). Figure 1(b) shows the drone energy consumption per unit traveling distance with varying payloads. The triangular symbols indicate optimal drone speeds that maximize the total traveling distance with fixed battery capacity, which can be observed more distinctly from Figure 1(d). It can be seen from Figure 1(b) that the drone energy consumption per unit distance initially decreases and subsequently rises again as the speed increases. This is because there is a balance between the flight duration and the energy consumption. Flying at lower speeds requires less power, which leads to a longer flight duration. In contrast, higher speeds result in more power being consumed, but the makespan is reduced. Therefore, the flight range notably depends on its speed, and faster flight speeds are not always optimal. Consequently, higher speeds can cause faster deliveries but with reduced range and increased energy consumption. This is essential within the scope of drone delivery systems because, in some cases, the customer is sufficiently remote from the warehouse, and flying at maximum speed can lead to a shorter range that falls short of the distance necessary to reach the customer. It may be beneficial to fly at lower speeds which can improve range and serve more customers. Moreover, speed adjustments can enhance endurance, which enables drones to serve more customers. As such, it is imperative to factor in drone speeds during deliveries when considering energy consumption.

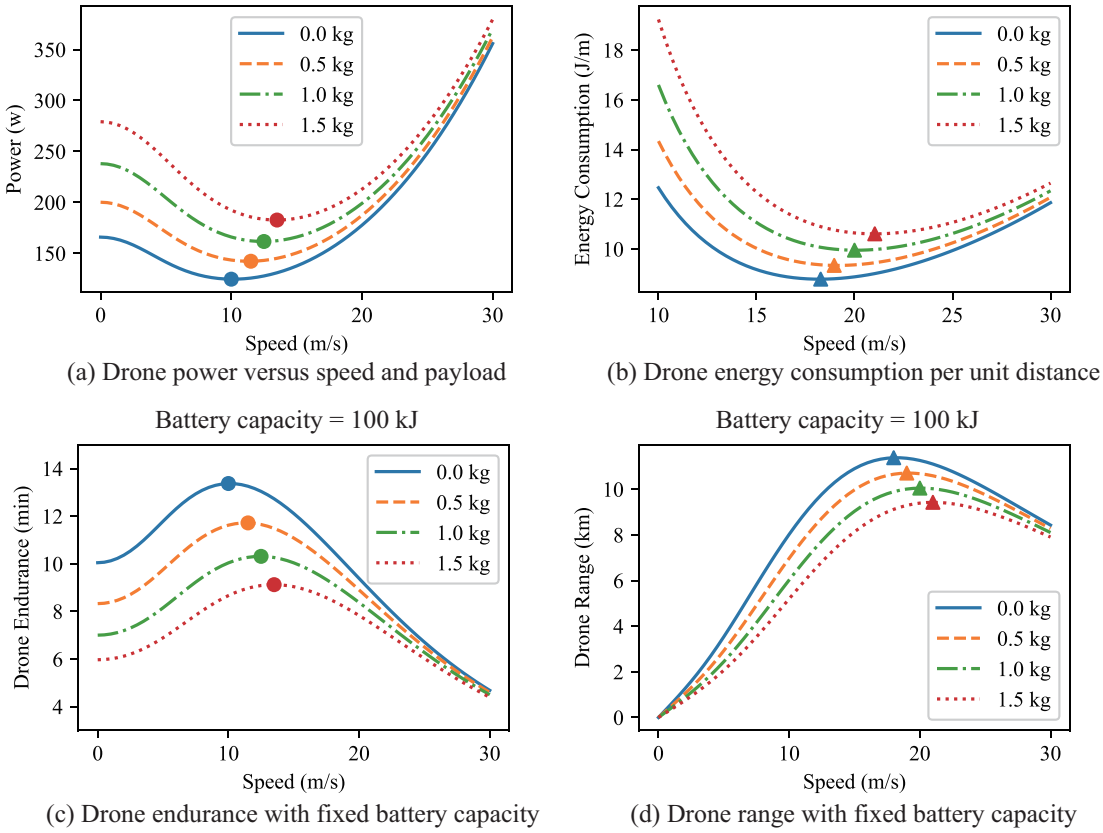
This paper introduces the multi-trip drone routing problem with variable flight speeds (DRP-VFS), an extension of the DRP proposed by [14]. In this problem, a fleet of homogeneous drones delivers multiple parcels to customers while minimizing energy consumption. In this case, drones can perform multiple trips and only be dispatched from and returned to the depot once per trip. While existing literature on the DRP assumes that drones fly at constant speeds in advance, the DRP-VFS considers drone speeds as decision variables when drones make deliveries.

This paper has the following contributions.

- We introduce and formally define a multi-trip drone routing problem, aiming to achieve operational cost minimization, where drone speeds are considered as decision variables rather than constants.
- We clearly consider drone energy consumption as a nonlinear function of both flight speed and payload rather than supposing that flight endurance and range are constants.
- We proposed a three-phased method integrating variable drone speed to solve DRP-VFS, which can improve the solution in terms of computational time to a greater extent than MILP.

The rest of this paper is structured as follows. In Section 2, we present a review of the literature. Section 3 presents the formal description of DRP-VFS and its MILP formulation. In Section 4, an efficient heuristic method proposed to solve large-scale instances is introduced. Section 5 demonstrates the case study and verifies the effectiveness of the proposed method. In the end, conclusions are drawn, and future research work is discussed in Section 6.

 doi:10.20944/preprints202403.1028.v1



(a) Drone power versus speed and payload

(b) Drone energy consumption per unit distance

(c) Drone endurance with fixed battery capacity

(d) Drone range with fixed battery capacity

Figure 1. The required power, energy consumption per unit distance, endurance, and range of a drone versus speed and payload referenced from [15].

2. Literature Review

Over the past years, there have been increasing studies on the civil application of drones ([4,16]). These studies reported that operational planning for drone delivery is a well-studied problem. There already exists a lot of literature on the vehicle routing problem (VRP) involving conventional delivery vehicles and its variants ([17,18]). However, adapting the VRP to drones is faced with additional challenges since delivery by drones must account for extra constraints such as lower battery capacity and higher sensitivity of energy cost to payload and flight speed ([6]).

Most drone studies have assumed either a fixed endurance or a constant speed ([10,19–24]). Obviously, under this assumption, drones flying at higher speed have longer ranges. Other studies consider a constant range of UAVs. These two assumptions are equivalent when drone speeds are constant. However, both of them fail to consider the influence of payload and speed on the drone's energy consumption, thus limiting the flexibility of drone delivery. For example, drones can extend their flight endurance by adjusting speed to serve customers further away from the depot.

Some studies have improved the fixed endurance/range model by considering drone power consumption dependent on payload while remaining unrelated to drone speed. [25] apply the UAV power rate function proposed by [26] and assume that the UAV maintains the maximum power during flight to calculate the load-dependent UAV speed during delivery. [27] propose a package delivery schedule utilizing drones and consider the drone's power rate varies linearly with its payload. [6] propose two drone routing problems where customers are served by drones only. The authors consider the power rate function linearly dependent on payload weight and develop a simulated annealing (SA) algorithm to address the problems. The power rate function adopted in [6] pertains solely to hovering without encompassing that of actual flight. [14] adopt the same energy consumption as in [6], but instead of approximating the power rate function, they develop an exact algorithm adding logical

 doi:10.20944/preprints202403.1028.v1

and subgradient cuts to solve a DRP with time windows. [28] use the UAV power rate proposed by [29] as a piecewise linear function of package weight in their DRP. [30] also consider the influence of package weight on energy consumption using the same model as [6]. However, different from the proposed model based on varying speeds, the studies in the above papers only consider constant drone speeds. Moreover, the proposed energy-consumption models are mainly about hovering and do not consider the energy cost of actual flight status. Other studies apply an energy consumption model related to the payload and the drone speed during forward flight. [31] adopt an energy model by [26], which is linearly dependent on payload and speed. [32] also consider the energy consumption function introduced in [11] in the multiple flying sidekicks traveling salesman problem (mFSTSP). However, the UAV speeds are fixed in these two papers, limiting the potential time and energy savings from flying at varying speeds.

Some recent studies have considered variable flight speeds in drone routing problems. For instance, [33] considers the reciprocal of the flight speed to be approximately linear with the weight of loads. However, they consider drone speed as a function of the load, which can be regarded as a fixed parameter related to given loads rather than as a decision variable to be optimized. [34] consider drone speeds as decision variables and use the energy model introduced by [12]. Besides, [35] employs the multi-rotor UAV energy consumption model developed by [11] and considers power rate as a function of a UAV's payload and speed. [35] investigate the mFSTSP with variable drone speeds. These two studies treat drone speeds as decision variables and demonstrate that operating drones at varying speeds contribute to delivery cost and time reductions, respectively. Moreover, [34] emphasizes that using fixed speeds may result in infeasibility in some instances. These two papers consider drone and truck tandem delivery, whereas our focus is solely on drone-only deliveries. As far as we know, this paper presents for the first time a DRP-VDS that considers UAV energy consumption and variable flight speeds.

Table 1 demonstrates a summary of relevant studies on the drone delivery problem reviewed in this paper. The studies are compared using the following factors: (i) the type of delivery (truck-drone tandem or drone-only), (ii) whether the endurance of drones is fixed or evaluated by energy consumption model, (iii) the drone speed (constant or variable), (iv) the objective function used, and (v) whether the solution methods are exact or heuristic. In the literature mentioned above, it is evident that only a few papers clearly consider the energy cost model and even fewer take drone speeds into account as decision variables. Because the finite drone range is susceptible to speed and payload weight, it is essential to consider variable drone speed and payload in drone delivery systems. In this study, a DRP-VFS is proposed to consider the energy constraints and variable speed and payload of the drone delivery systems for last-mile delivery systems. Also, a three-phased heuristic method is proposed to solve large-scale scenarios. The performance of the proposed heuristic is evaluated by comparing it to the exact method.

 doi:10.20944/preprints202403.1028.v1

Table 1. Summary of studies on drone routing problem in the literature.

Reference	Delivery type		Energy model	Drone Speed	Objective	Solution method
	Truck&Drone	Drone				
[19]	✓			Constant	Delivery time	Exact & Heuristic
[20]	✓			Constant	Delivery time	Heuristic
[21]	✓			Constant	Delivery time	–
[22]	✓			Constant	Operational cost	Heuristic
[23]	✓			Constant	Operational cost	Heuristic
[10]	✓			Constant	Delivery time	Exact & Heuristic
[24]	✓			Constant	Delivery time	Exact
[25]	✓		[26]	Constant	Vehicle waiting time	Heuristic
[27]		✓	Hovering	Constant	Drone fleet size	Exact & Heuristic
[6]		✓	Hovering	Constant	Operational cost	Exact & Heuristic
[14]		✓	Hovering	Constant	Operational cost	Exact
[28]	✓		[29]	Constant	Delivery time	Heuristic
[30]	✓		Hovering	Constant	Delivery time	Exact & Heuristic
[31]		✓	[26]	Constant	Operational cost	Exact
[32]	✓		[11]	Constant	Delivery time	Exact & Heuristic
[33]	✓		Linear	Variable	Delivery time	Exact
[34]	✓		[12]	Variable	Operational cost	Exact & Heuristic
[35]	✓		[11]	Variable	Delivery time	Heuristic
This paper		✓	[12]	Variable	Operational cost	Exact & Heuristic

3. Model Formulation

This section presents a formulation of the DRP-VFS, and introduces piecewise linearization to approximate the nonlinear energy consumption function.

3.1. Problem Definition

The multi-trip drone routing problem with variable flight speeds (DRP-VFS) is defined on a complete directed graph $G = (N, A)$, where $N = \{0, 1, \ldots, n+1\}$ represents the set of all nodes with the node 0 and $n+1$ corresponding to the depot (distribution center) from which all drones must originate and return. The set of customers is denoted by $N' = \{1, \ldots, n\}$ associated with the distinct parcels. To simplify notation, we introduce $N^+ = \{0, \ldots, n\}$ to indicate the set of nodes from which a drone can depart and $N^- = \{1, \ldots, n+1\}$ to describe the set of nodes that a drone may visit. Let

 doi:10.20944/preprints202403.1028.v1

$A = \{(i,j) : i \in \{0\}, j \in N' \text{ and } i \in N', j \in N^-, i \neq j\}$ denote the set of arcs. Additionally, we define sets $\delta^-(i)$ and $\delta^+(i)$ to denote the forward and backward nodes of node i, respectively.

Each customer has a positive demand q_i and a hard time window $[a_i, b_i]$. The drones are allowed to arrive early but need to wait until a_i to begin serving the customer. A fleet of U homogeneous multi-rotor drones departs from the depot with a fully charged battery. Q represents the drone's maximum payload, and we suppose that $q_i \leq Q, \forall i \in N'$, while E is the maximum energy provided by a fully charged battery. The service time for each customer is s_i. Every drone can implement multiple trips and serve several customers within one delivery. d_{ij} represents flight distance on arc (i,j), which is assumed to be symmetric. Since the drone speeds are decision variables in this paper, the flight time on arc (i,j) is non-constant, as discussed later. The maximum speed limit is represented by V_{max}.

We make the following assumptions:

- Parcels are delivered by drones only.
- There is only one depot from which drones can depart.
- We neglect the time of loading parcels and swapping batteries.
- Without loss of generality, we ignore the influence of weather, i.e., wind impact is not considered.
- Drones can fly at a constant speed between two locations. The speed of each flight can vary.

3.2. Mathematical Model

DRP-VFS involves two sets of binary variables: $x_{ij} = 1$ if a drone flies across arc (i,j), and $x_{ij} = 0$ otherwise. $z_{ij} = 1$ if a trip ends in node i and is immediately succeeded by another trip with node j as the first customer. Additionally, there are six sets of continuous variables: m_{ij} is the parcel's weight transported across the arc (i,j) (kg). v_{ij} denotes the drone speed from node $i \in N^+$ to node $j \in N^-$ (m/s). τ_i is the service start time of customer $i \in N^-$ (second). t_{ij} represents the time required to fly through arc (i,j). f_i denotes the total energy a drone consumes when it arrives at customer i (kWh). e_{ij} indicates the energy consumed through arc (i,j), measured in kWh.

The constraints of route feasibility can be expressed as follows:

$$\sum_{j \in \delta^+(i)} x_{ij} = 1 \quad \forall i \in N', \tag{1}$$

$$\sum_{j \in \delta^-(i)} x_{ji} = 1 \quad \forall i \in N', \tag{2}$$

$$\sum_{j \in N'} x_{0j} = \sum_{j \in N'} x_{j,n+1}. \tag{3}$$

Constraints (1) and (2) guarantee that each customer is served only once. Constraint (3) guarantees that the number of trips leaving the depot equals the number returning to the depot.

We apply demand constraints through

$$\sum_{j \in \delta^-(i)} m_{ji} - \sum_{j \in \delta^+(i)} m_{ij} = q_i \quad \forall i \in N', \tag{4}$$

$$m_{i,n+1} = 0 \quad \forall i \in N', \tag{5}$$

$$m_{ij} \leq Q x_{ij} \quad \forall (i,j) \in A. \tag{6}$$

Constraints (4) ensure that the demand of each customer can be satisfied, and also eliminate subtours. Consequently, the payload decreases to zero when the drone returns to the depot as specified by constraints (5). With constraints (6), the total weight of parcels carried by a drone in a trip must not exceed its payload capacity.

 doi:10.20944/preprints202403.1028.v1

The following equations are formulated to enforce time-related constraints.

$$\tau_i + s_i + d_{ij}/v_{ij} \leq M(1 - x_{ij}) + \tau_j \quad \forall i \in N', j \in N^-, \tag{7}$$

$$a_i \leq \tau_i \leq b_i \quad \forall i \in N^-, \tag{8}$$

$$\tau_i + s_i + (t_{i,n+1} + t_{0j}) \leq M(1 - z_{ij}) + \tau_j \quad \forall i, j \in N', i \neq j. \tag{9}$$

Constraints (7) require that the total of the arrival time and service time in i and travel time across the arc (i, j) not greater than the arrival time at the customer j (the immediate successor of i). Constraints (8) indicate that the arrival time to customer i falls within its designated time window $[a_i, b_i]$. Constraints (9) indicate the temporal relationship between successive trips executed by one drone, including the return time. Note that constraints (7) are nonlinear due to $\frac{1}{v_{ij}}$, we will introduce a piecewise linear method to convert them into linear ones in subsection 3.3.

In this paper, we allow drones to change batteries and place parcels for new trips when they return to the depot. We adopt a 2-index formulation inspired by [14] to formulate the multi-trip characteristics of drone delivery. Therefore, we can achieve the reusability of drones by incorporating the following constraints:

$$\sum_{\substack{i \in N' \\ i \neq j}} z_{ij} \leq x_{0j} \quad j \in N', \tag{10}$$

$$\sum_{\substack{j \in N' \\ j \neq i}} z_{ij} \leq x_{i,n+1} \quad i \in N', \tag{11}$$

$$\sum_{j \in N'} x_{0j} - \sum_{i \in N'} \sum_{\substack{j \in N' \\ j \neq i}} z_{ij} \leq U. \tag{12}$$

Constraints (10) and (11) establish a relationship between variables $\mathbf{x}$ and $\mathbf{z}$. Constraint (12) limits the number of drones that can be used.

To evaluate the energy usage during delivery, we adopt the power rate model according to [12].

$$P(v, w) = \frac{\sigma}{8}\rho s D \Omega^2 R^3 (1 + \frac{3v^2}{U_{tip}^2}) + \frac{\kappa w^{\frac{3}{2}}}{\sqrt{2\rho D}} \left(\sqrt{\frac{v^4}{4v_0^4} + 1} - \frac{v^2}{2v_0^2} \right)^{\frac{1}{2}} + \frac{1}{2}\rho S_{FP} v^3. \tag{13}$$

where σ is the average profile drag coefficient, ρ is the air density, s is the rotor solidity, Ω is the angular velocity of the rotor, U_{tip} is the blade tip speed, R is the rotor radius, D is the rotor disc area, κ is the correction factor to ideal induced power, v_0 is the induced velocity in hovering, S_{FP} is the equivalent flat plate area of the drone. w is the total weight of the drone, including the unloaded weight m_v of the drone and the weight m_p of parcels. Thus, $w = (m_v + m_p)g$ on the forward flight to customers, where g is the gravitational acceleration. To simplify the above formula, define $P_0 = \frac{\sigma}{8}\rho s D \Omega^2 R^3$ and $P_i = \frac{\kappa w^{\frac{3}{2}}}{\sqrt{2\rho D}}$ as the blade profile power and induced power when hovering.

By transforming (13), we derive an energy consumption formula denoted as $E_0(v, m)$ (J/m) for a drone's travel over per unit distance at a constant speed v, and calculated as

$$E_0(v, w) = \frac{P(v, w)}{v} = P_0 \left(\frac{1}{v} + \frac{3v}{U_{tip}^2} \right) + P_i \left(\sqrt{v^{-4} + \frac{1}{4v_0^4}} - \frac{1}{2v_0^2} \right)^{\frac{1}{2}} + \frac{1}{2}\rho S_{FP} v^2. \tag{14}$$

Thus, the energy consumption for drone on arc (i, j), denoted as e_{ij} can be expressed as

$$e_{ij} = \begin{cases} E_0(v_{ij}, w_{ij}) \cdot d_{ij} & x_{ij} = 1, \\ 0 & \text{otherwise}, \end{cases} \tag{15}$$

 doi:10.20944/preprints202403.1028.v1

where d_{ij} and $w_{ij} = (m_v + m_{ij})g$ are the distance and payload on arc (i, j). It is worth noting that (15) is nonlinear when $x_{ij} = 1$ and a piecewise linear method will be applied to deal with them in the next subsection.

Hence, drones' energy constraints are written as

$$f_0 = 0, \tag{16}$$
$$f_i + e_{ij} \leq M(1 - x_{ij}) + f_j \quad \forall (i, j) \in A, \tag{17}$$
$$f_{n+1} \leq E. \tag{18}$$

Constraint (16) signifies that the initial energy consumption of each trip is set to zero. Constraints (17) indicate the energy consumed across the arc (i, j). Constraint (18) guarantees the battery's energy capacity constraint must be imposed.

The domain of the variables is shown below.

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in A, \tag{19}$$
$$z_{ij} \in \{0, 1\} \qquad \forall i, j \in N', \tag{20}$$
$$m_{ij}, e_{ij}, t_{ij} \geq 0 \quad \forall (i, j) \in A, \tag{21}$$
$$0 \leq v_{ij} \leq V_{max} \quad \forall (i, j) \in A, \tag{22}$$
$$f_i \geq 0 \qquad \forall i \in N, \tag{23}$$
$$\tau_i \geq 0 \qquad \forall i \in N^-. \tag{24}$$

3.3. Piecewise Linearization

Piecewise linearization can approximate a nonlinear univariate function using a set of non-negative variables of special ordered sets type 2 (SOS2), where at most two adjacent variables can be positive, while the sum of all variables equals 1 ([36]). For univariate functions like $1/v$, the piecewise linearization approximation is realized by sampling K points $[V_1, V_2, \ldots, V_K]$ as breakpoints on the v-axis. After that, the linear segments $[(V_1, 1/V_1), \ldots, (V_K, 1/V_K)]$ are used to approximate $1/v$.

In this case, we define a set of continuous variable α_{ij}^k for each breakpoint. The value of v_{ij} on arc (i, j) is represented by

$$v_{ij} = \sum_{k=1}^{K} V_k \alpha_{ij}^k. \tag{25}$$

The approximation value of the reciprocal of speed $1/v_{ij}$ is $1/\tilde{v}_{ij}$ which can be calculated by (26).

$$\frac{1}{\tilde{v}_{ij}} = \sum_{k=1}^{K} \frac{\alpha_{ij}^k}{V_k}. \tag{26}$$

The SOS2 variables α_{ij}^k need to satisfy the following constraints.

$$\sum_{k=1}^{K} \alpha_{ij}^k = 1, \tag{27}$$
$$0 \leq \alpha_{ij}^k \leq 1 \quad k = 1, 2, \ldots, K. \tag{28}$$

Most modern MIP solvers, such as Gurobi and CPLEX, can address specially ordered sets of types 1 and 2 automatically. Consequently, constraints (25)–(28) are sufficient to produce the correct computation. Consequently, the nonlinear constraints in (7) are approximated by replacing $1/v_{ij}$ with $1/\tilde{v}_{ij}$.

Similarly, we introduce a set of SOS3 variables β_{ij}^{st} to linearly approximate the two-variable energy consumption function dependent on drone speed and parcel weight in (14). Similar to the

 doi:10.20944/preprints202403.1028.v1

SOS2, the SOS3 variables require that at most three adjacent components be greater than 0. The difference is that the set of SOS2 variables is one-dimensional, while the set of SOS3 variables is two-dimensional. Consider again S sampling coordinates $[V_1, V_2, \ldots, V_S]$ on the v-axis and T sampling coordinates $[M_1, M_2, \ldots, M_T]$ on the m-axis. The function $E_0(v, m)$ is determined by each breakpoint $(V_s, M_t)(s = 1, \ldots, S; t = 1, \ldots, T)$.

In our problem, the value of v_{ij} and m_{ij} on arc (i, j) are represented by

$$v_{ij} = \sum_{s=1}^{S} V_s \beta_{ij}^{st}, \tag{29}$$

$$m_{ij} = \sum_{t=1}^{T} M_t \beta_{ij}^{st}. \tag{30}$$

The approximation value of the energy consumption is $\tilde{e}_{ij}$ which can be calculated by using (31).

$$\tilde{e}_{ij} = \sum_{s=1}^{S} \sum_{t=1}^{T} E_0(V_s, M_t) \beta_{ij}^{st}. \tag{31}$$

The SOS3 variables β_{ij}^{st} should satisfy (32) and (33).

$$\sum_{s=1}^{S} \sum_{t=1}^{T} \beta_{ij}^{st} = 1, \tag{32}$$

$$0 \leq \beta_{ij}^{st} \leq 1 \quad s = 1, 2, \ldots, S, t = 1, 2, \ldots, T. \tag{33}$$

However, unlike SOS1 and SOS2, modern MIP solvers don't have an automatic syntax for imposing SOS3. For this reason, we utilize the triangle method proposed by [37] to impose corresponding constraints. Consider the rectangle corresponding to intervals $[V_s, V_{s+1})$ and $[M_t, M_{t+1})$ and the two triangles generated by its diagonal $[(V_s, M_t)(V_{s+1}, M_{t+1})]$. We introduce two sets of binary variables u_{ij}^{st} and l_{ij}^{st} to represent the upper and lower triangles in the rectangle, respectively, with dummy values $u_{ij}^{0*} = u_{ij}^{*0} = u_{ij}^{S*} = u_{ij}^{*T} = 0$ and $l_{ij}^{0*} = l_{ij}^{*0} = l_{ij}^{S*} = l_{ij}^{*T} = 0$ at the boundaries. The additional constraints are shown as follow:

$$\sum_{s=1}^{S-1} \sum_{t=1}^{T-1} \left(u_{ij}^{st} + l_{ij}^{st} \right) = 1, \tag{34}$$

$$\beta_{ij}^{st} \leq u_{ij}^{st} + l_{ij}^{st} + u_{ij}^{s,t-1} + l_{ij}^{s-1,t-1} + u_{ij}^{s-1,t-1} + l_{ij}^{s-1,t}. \tag{35}$$

The constraint (34) requires that just one triangle be chosen from all possible triangles. Constraints (35) guarantee that the only β_{ij}^{st} values greater than 0 can be those corresponding to the three vertices of such a triangle. Accordingly, the nonlinear energy-related constraints in (17) are approximated by replacing e_{ij} with $\tilde{e}_{ij}$.

The objective function is minimizing the total energy consumption during delivery.

$$\begin{aligned} &\min \sum_{(i,j) \in A} e_{ij}, \\ &\text{s.t. } (1) - (35). \end{aligned} \tag{36}$$

The model described above is a mixed-integer linear programming (MILP) model which can be solved efficiently by commercial MIP solvers such as Gurobi. While MILP solvers find optimal solutions, the computational time required is substantial, even for scenarios with several customers. In Section 4, we propose a three-phased heuristic algorithm aimed at finding suboptimal solutions within computational time constraints.

 doi:10.20944/preprints202403.1028.v1

4. Solution Method

The DRP-VFS extends the classical VRP and is an NP-hard problem. Owing to the NP-hardness of the DRP-VFS, a three-phased heuristic is proposed to effectively tackle practical large-scale scenarios.

The solution of DRP-VFS can be represented by $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_\mathcal{K}\}$, where $\mathcal{K}$ is the number of trips. The $\mathbf{r}_k$ contains a vector of customers that a drone will visit, which can be expressed as $\mathbf{r}_k = \{r_1, r_2, \ldots, r_{C_k}\}$, where C_k is the number of customers assigned to the kth trip.

4.1. Initialization

There are numerous heuristic approaches that can efficiently search for a feasible solution for the VRP, such as CW savings algorithm and sweep algorithm. In our method, we first apply an improved k-means algorithm that includes capacity and Euclidean distance as constraints to partition customers into several subsets (Algorithm 1). Next, a traveling salesman problem with time windows (TSPTW) is solved within each subset to determine the customer sequence a drone serves. This algorithm is executed while satisfying the capacity, energy, and time window constraints. The number of clusters is calculated by the total demands and capacity of the drone as

$$\mathcal{K} = \lceil \sum_{i=1}^{n} q_i / Q \rceil \tag{37}$$

The $\mathcal{K}$ centroids are randomly initialized in the customers' coordinate range. The Euclidean distances between each customer and all $\mathcal{K}$ centroids are calculated. Sort customers by demand from largest to smallest and assign each customer to the nearest centroids. The total demands for each subset of clusters cannot exceed the drone's capacity. The selected customer will be assigned to the next nearest centroid if the capacity constraint is not satisfied. The centroid of kth cluster (x_k, y_k) is calculated based on its members using (38).

$$\begin{aligned} x_k &= \sum_{i=1}^{C_k} x_i / C_k, \\ y_k &= \sum_{i=1}^{C_k} y_i / C_k. \end{aligned} \tag{38}$$

The iterative procedure is repeated until the cluster is unchanged or the maximum iteration is reached.

Algorithm 1 K-means Clustering.

Require: a set of customers $R = \{r_1, r_2, \ldots, r_n\}$.
1: Initialization: calculate $\mathcal{K}$ using (37) and initialize the centroid randomly. Sort customers by demand.
2: **while** not converged **do**
3: **for** each customer $r_i \in R$ **do**
4: Calculate the distance from r_i to each of the $\mathcal{K}$ clusters and sort it in descending order.
5: **while** r_i is not assigned **do**
6: Assign r_i to the nearest centroid without violating the capacity constraint.
7: **if** r_i is not assigned **then**
8: Choose the next nearest centroid.
9: **end if**
10: **end while**
11: **end for**
12: Calculate the new centroid of the clusters using (38).
13: **end while**
Ensure: $\mathcal{K}$ clusters

 doi:10.20944/preprints202403.1028.v1

The TSPTW is a special case of DRP-VFS since only one drone is used. Thus, we just need to remove the reusability constraints (10)-(12) and modify constraints (3) to the following.

$$\sum_{r\in\mathbf{r}_k} x_{0r} = \sum_{r\in\mathbf{r}_k} x_{r,n+1} = 1 \quad k = 1,2,\ldots,\mathcal{K}. \tag{39}$$

Therefore, an initial feasible solution can be obtained quickly.

4.2. Local Search

In this section, we propose a speed optimization model to adjust the flight speed in a trip and calculate the cost of a DRP-VFS solution, and the Adaptive Large Neighborhood Search (ALNS) algorithm is used to enhance the solution quality.

4.2.1. Speed Optimization

The speed optimization problem (SOP) involves adjusting the flight speed in order to minimize energy consumption. The SOP is defined on a route $\mathbf{r} = \{r_0, r_1, \ldots, r_{C+1}\}$ consisting of customers served by a single drone, where r_0 and r_{C+1} represent the depot corresponding to 0 and $n+1$, respectively. v_i and m_i denote the flight speed and payload carried by a drone between node r_i and its successor. The SOP is formulated as follows:

$$\min \sum_{i=0}^{C} e_i, \tag{40}$$

subject to

$$m_i = \sum_{j=i+1}^{C} q_j \quad i = 0,\ldots,C, \tag{41}$$

$$m_0 \le Q, \tag{42}$$

$$\tau_i + s_i + d_i/v_i \le \tau_{i+1} \quad i = 0,\ldots,C, \tag{43}$$

$$a_i \le \tau_i \le b_i \quad i = 1,\ldots,C, \tag{44}$$

$$e_i = E_0(v_i, m_i)d_i \quad i = 0,\ldots,C, \tag{45}$$

$$f_0 = 0, \tag{46}$$

$$f_i + e_i \le f_{i+1} \quad i = 0,\ldots,C, \tag{47}$$

$$f_{C+1} \le E. \tag{48}$$

The nonlinear energy function is linearized using SOS2 variables introduced in section 3.3. Since m_i is calculated in advance using (42), the E_0 between node r_i and its successor depends only on v_i. We introduce γ_i^s to linearly approximate the energy consumption function. Hence, the value of v_i and the approximate value of $\tilde{e}_i$ can be represented by

$$v_i = \sum_{s=1}^{S} V_s \gamma_i^s, \tag{49}$$

$$\tilde{e}_i = \sum_{s=1}^{S} E_0(V_s, m_i)\gamma_i^s. \tag{50}$$

To deal with $1/v_i$, the same method can be used. However, to speed up the solving process, we introduce a convexification method, i.e., replace $1/v_i$ with an auxiliary variable λ_i to avoid reintroducing integer variables:

$$\tau_i + s_i + d_i\lambda_i \le \tau_{i+1}. \tag{51}$$

 doi:10.20944/preprints202403.1028.v1

Meanwhile, the following inequality should be imposed to ensure λ_i converges to $1/v_i$.

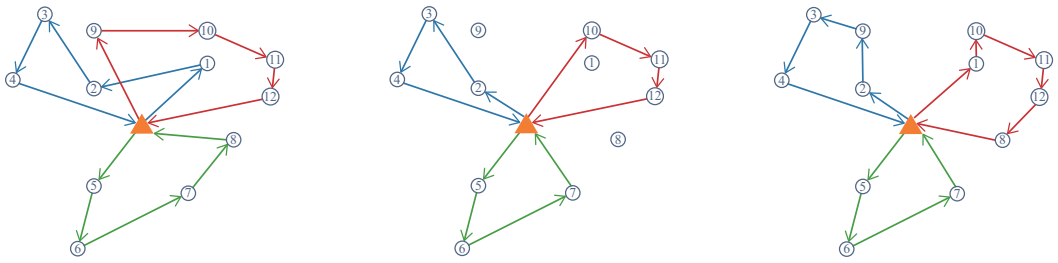
$$\lambda_i \geq 1/v_i. \tag{52}$$

It can be seen that constraint (52) is a quadratic and convex constraint. Therefore, the SOP is modeled as a mixed-integer quadratically constrained programming (MIQCP) and can be solved by commercial solvers.

4.2.2. ALNS heuristics

In this subsection, an adaptive large neighborhood search algorithm is adopted to improve the initial solution obtained by Section 4.1. The ALNS framework has demonstrated successful results in solving multiple variants of vehicle routing problems ([38]). By removing selected nodes from the solution and reinserting them, we obtain a neighborhood of the given solution. The operators are dynamically chosen based on their previous performance. For this reason, each operator is given a score that increases if it enhances the current solution. A new solution is accepted if it meets the criteria specified by the local search scheme, such as simulated annealing.

In this paper, we apply two removal operators and two insertion operators to effectively destroy and repair a DRP-VFS solution. In addition, for every Γ iteration, the TSPTW model is solved once within each trip to adjust the customer sequence. The framework of the ALNS algorithm is presented in Algorithm 2. Figures 2(a)-(c) depict the initial solution, the solution after applying two destroy operators and the solution after executing two repair operators.



(a) Initial DRP-VFS solution

(b) Destroy operators: blue route removal {1}, red route removal {9}, green route removal {8}

(c) Repair operators: blue route inseration {9}, red route inseration {1, 8}

Figure 2. Example of ALNS heuristic on DRP-VFS solution.

The destroy operators involve primarily removing several nodes from the current solution and placing them in a removal list $\mathcal{L}$. We present the removal operators involved in our approach as follows:

(1) Random removal: This operator removes several nodes at random from the current solution.
(2) Worst removal: This operator removes the highest-cost node from the solution, where the cost is determined by solving SOP.

The insertion operators are applied to repair a partly destroyed solution by reinserting the nodes from $\mathcal{L}$ back into the solution. The insertion operators employed in our algorithm include:

(1) Greedy insertion: This operator repeatedly removes a node from $\mathcal{L}$ and inserts it into the lowest cost position of a route.
(2) Regret insertion: An obvious disadvantage of greedy insertion is that it defers node insertion to later iterations where few feasible moves are available. The regret operator in our algorithm uses a 2-regret criteria. Define Δc_i^j as the cost change incurred by inserting node i into the route where the cost is jth cheapest. The 2-regret criteria inserts the node i based on $i^* = \underset{i \in \mathcal{L}}{\operatorname{argmax}} \left(\Delta c_i^2 - \Delta c_i^1\right)$, where Δc_i^1 and Δc_i^2 are the best and second-best insertion of node i. We iterate the procedure until no more nodes in $\mathcal{L}$ can be inserted.

 doi:10.20944/preprints202403.1028.v1

We describe the adaptive weight adjustment process below. The operators are chosen by the roulette-wheel method. In the beginning, all operators are equally likely. They are updated as $P_i^{t+1} = P_i^t(1-\Theta) + \Theta\pi_i/\omega_i$ during program execution, where P_i^t is the probability of operator i at the tth iteration, Θ is the parameter of roulette wheel, π_i denotes the score of operator i and ω_i indicates times it was selected in the previous Γ iterations. The score reflects the performance of each operator. When a new best solution is found, the related operators' scores are increased by σ_1. If the new solution outperforms the current one, the score is increased by σ_2. If the new solution falls short of the current one but meets the acceptance criteria, the score is increased by σ_3.

Our ALNS algorithm applies the simulated annealing as an acceptance criterion. We calculate the probability in line 12 to determine whether to update the current solution and decrease the initial temperature T according to $T = \Phi T$, where Φ is a constant indicating the cooling rate. The algorithm produces the best solution that has been found after Λ iterations.

Algorithm 2 ALNS with simulated annealing.

Require: Removal operators $\mathcal{D}$, insertion operators $\mathcal{I}$, cooling rate Φ.
1: Produce an initial solution using the K-means algorithm in section (4.1).
2: Initialize temperature T and counter $t \leftarrow 1$.
3: Initialize probability P_d^t for each destroy operator and probability P_i^t for each repair operator.
4: Let $\mathcal{R}_{current} \leftarrow \mathcal{R}_{best} \leftarrow \mathcal{R}_{init}$.
5: **while** $t \leq \Lambda$ **do**
6: Select a removal operator $d^* \in \mathcal{D}$ and a insertion operator $i^* \in \mathcal{I}$.
7: Apply operator d^* to $\mathcal{R}_{current}$ to generate $\mathcal{R}'_{new}$.
8: Apply operator i^* to $\mathcal{R}'_{new}$ to generate $\mathcal{R}_{new}$.
9: **if** $c(\mathcal{R}_{new}) < c(\mathcal{R}_{current})$ **then**
10: $\mathcal{R}_{current} \leftarrow \mathcal{R}_{new}$
11: **else**
12: Let $\nu \leftarrow e^{-(c(\mathcal{R}_{new}) - c(\mathcal{R}_{current})/T)}$
13: **end if**
14: Get a random number $\xi \in [0, 1]$
15: **if** $\xi \leq \nu$ **then**
16: $\mathcal{R}_{current} \leftarrow \mathcal{R}_{new}$
17: **end if**
18: **if** $c(\mathcal{R}_{current}) < c(\mathcal{R}_{best})$ **then**
19: $\mathcal{R}_{best} \leftarrow \mathcal{R}_{current}$
20: **end if**
21: $T \leftarrow \Phi T$
22: **if** $t \bmod \Gamma = 0$ **then**
23: Apply the adaptive weight adjustment procedure to update probabilities.
24: Solve TSPTW model within each route to adjust the customer sequence.
25: **end if**
26: $t \leftarrow t + 1$
27: **end while**
Ensure: $\mathcal{R}_{best}$

4.3. Assignment

In this section, we describe a list schedule algorithm to minimize the number of drones. The delivery time, denoted by h, is the latest allowable departure time from the depot, whereas the arrival time, denoted by l, indicates when the drone reaches the depot upon finishing the delivery. The two times are calculated by using (53).

$$\begin{aligned} h &= \tau_1 - d_0/v_0, \\ l &= \tau_C + s_C + d_C/v_C, \end{aligned} \tag{53}$$

where τ_1 and τ_C are the starts of service time at the first and last nodes in a route, d_0, v_0 and d_C, v_C is the distance and speed at the first and last flights in a route, and s_C is the service time of the last node.

 doi:10.20944/preprints202403.1028.v1

We allocate $\mathcal{K}$ routes to U drones using the list scheduling algorithm (Algorithm 3). First, sort routes by delivery time in ascending order. A binary heap is used to get the route with minimum arrival time quickly. Next, traverse the $\mathcal{K}$ routes; if the delivery time of $\mathbf{r}_k$ is greater or equal to the arrival time of the heap top route, then $\mathbf{r}_k$ is appended to this route. Otherwise, a new drone should be added. The time complexity is $\mathcal{O}(\mathcal{K} \log \mathcal{K})$. In conclusion, our heuristic method finds a near-optimal solution $\mathcal{R}^*$ for DRP-VFS.

Algorithm 3 List Schedule.

Require: a solution $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_\mathcal{K}\}$, the number of drones U.
Initialize near-optimal solution $\mathcal{R}^* \leftarrow \emptyset$
Initialize binary heap $\mathcal{Q} \leftarrow \emptyset$
for $i \in \{1, \ldots, U\}$ **do**
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup (\emptyset, 0)$
end for
Sort $\mathcal{R}$ by delivery time in ascending order.
for $k \in \{1, \ldots, \mathcal{K}\}$ **do**
  $(\mathbf{r}, l) \leftarrow$ POPMINIMUMARRIVALTIMEELEMENT$(\mathcal{Q})$
  if $l \leq h_k$ **then**
    $\mathbf{r} \leftarrow \mathbf{r} \cup \mathbf{r}_k$
    $l \leftarrow l_k$
  else
    PUSHELEMENT$(\mathcal{Q}, (\mathbf{r}_k, l_k))$
  end if
  PUSHELEMENT$(\mathcal{Q}, (\mathbf{r}, l))$
end for
while $\mathcal{Q}$ is not empty **do**
  $\mathbf{r} \leftarrow$ POPELEMENT$(\mathcal{Q})$
  $\mathcal{R}^* \leftarrow \mathcal{R}^* \cup \mathbf{r}$
end while
Ensure: $\mathcal{R}^*$.

5. Results and Discussion

In this section, numerical experiments have been carried out to verify the computational efficiency of the proposed heuristic method for the DRP-VFS. Firstly, the computational performance of the ALNS algorithm is revealed by comparing its solution quality and CPU time to those of the exact method with small-scale scenarios in section 5.2. Secondly, the performance of ALNS implementation in large-scale instances is demonstrated in section 5.3. Finally, we compare the solutions generated by fixed-speed and variable-speed in section 5.4. It is demonstrated that the energy consumption of variable-speed solutions decreases notably compared to that of fixed-speed solutions. All computational work was conducted on the computer configuration with an Intel i7 CPU and 16 GB RAM. The heuristic algorithm was coded in Python 3.7 and Gurobi 9.5.0 was used to solve the mathematical models.

5.1. Parameter Settings

We use two sets of benchmark instances introduced by [14], named *Set A* and *Set B*. The *Set A* is created based on [39] and [6], which contains 10-50 customers. The *Set B* is an extension of Solomon's instance, which contains 10-100 customers. We use *Set A* to validate the effectiveness of MILP and ALNS implementation and energy savings with variable flight speeds. We use *Set B* to validate the efficiency of ALNS implementation on large-scale instances. We execute the MILP implementation once for each instance. We execute the ALNS algorithm 10 times for each instance and calculate the average objective value and computational time. The parameter settings of drones and the ALNS algorithm are shown in Table 2 and Table 3. For *Set A* instances, the battery energy capacity is set to $E = 0.27$ kWh; For *Set B* instances, we set $E = 0.027$ kWh.

 doi:10.20944/preprints202403.1028.v1

Table 2. Parameters of typical values for drones.

Notation	Description	Values
m_v	Mass of drone (kg)	2
s	Rotor solidity	0.05
v_0	Induced velocity in hover	4.03
κ	correction factor	1.1
ρ	Air density (kg/m^3)	1.225
σ	Profile drag coefficient	0.012
Ω	Angular velocity of the rotor (rad/s)	300
D	Rotor disc area in m^2, $D = \pi R^2$	0.503
R	Rotor radius (m)	0.4
S_{FP}	Equivalent flat plate area (m^2)	0.0151
U_{tip}	Blade tip speed (m/s)	120

Table 3. Parameters settings of ALNS algorithm.

Parameter	Λ	Γ	Θ	Φ	T	σ_1	σ_2	σ_3
Setting	1000	50	0.4	0.9	30	30	20	10

5.2. Performance Comparison between MILP and ALNS Implementations

In this section, we compare the performance of the MILP implementation with the ALNS implementation to demonstrate that the ALNS algorithm is effective in finding near-optimal solutions to small DRP-VFS instances. The results of MILP and ALNS are presented in Table 4. CPU is the computational time taken to solve the problem. The last three columns in Table 4 are the percentage differences of ALNS versus MILP in terms of energy consumption, total flight distance, and computational time. The reason why ALNS needs more drones than MILP is that the time window is not fully utilized when calculating the delivery time and arrival time using (53). For example, when appending $\mathbf{r}_1$ to $\mathbf{r}_2$, the delivery time of $\mathbf{r}_2$ is possible to postpone backward while satisfying the time window constraint. Nevertheless, ALNS does not take this scenario into account due to the challenges associated with its implementation.

We can observe that the CPU time of MILP grows exponentially with the number of customers. For instance, the average CPU time increases from 144.2 s to 666.9 s, up by 362.5 % when solving the 20-customer problem compared to the 15-customer problem. The average CPU time of the ALNS increases from 123.5 s to 149.4 s, a growth of 21.3 %. Moreover, for the first two instances with 10 customers, ALNS finds the optimal solution. It can be obviously found from Table 4 that the difference in energy consumption and travel distance between ALNS and MILP is less than 10 %, which reveals that ALNS is capable of finding near-optimal solutions in small scenarios.

 doi:10.20944/preprints202403.1028.v1

Table 4. Comparison of MILP and ALNS for Set A instances with size 10-20.

Cust	Inst	MILP				ALNS				Gap (%)		
		Energy	Distance	CPU	UAVs *	Energy	Distance	CPU	UAVs *	Energy	Distance	CPU
10	1	0.0133	4.475	19.5	2	0.0133	4.475	83.9	2	0.00	0.00	330.26
	2	0.0163	5.447	19.9	2	0.0163	5.447	80.4	2	0.00	0.00	304.02
	3	0.0137	4.582	18.4	2	0.0146	4.981	99.6	2.5	6.57	8.71	441.30
	4	0.0156	5.281	77.9	2	0.0157	5.320	110.9	2.1	0.64	0.74	42.36
	5	0.0144	4.845	52.5	2	0.0144	4.845	84.7	3	0.00	0.00	61.33
15	1	0.018	6.025	124.8	3	0.0184	6.121	127.7	4.6	2.22	1.59	2.32
	2	0.0221	7.382	228.9	3	0.0225	7.513	134.6	3.8	1.81	1.77	-41.20
	3	0.0183	6.042	93.4	4	0.0186	6.140	102.4	4	1.64	1.62	9.64
	4	0.0213	7.195	112.2	3	0.0227	7.515	137.3	3.8	6.57	4.45	22.37
	5	0.0231	7.74	161.6	3	0.0232	7.760	115.4	3.3	0.43	0.26	-28.59
20	1	0.0306	10.208	658.7	4	0.0307	10.211	130.2	4	0.33	0.03	-80.23
	2	0.0262	8.638	586.1	4	0.0270	9.026	158.4	4	3.05	4.49	-72.97
	3	0.0281	9.345	1039.2	4	0.0282	9.453	186.1	4	0.36	1.16	-82.09
	4	0.0266	8.819	495.1	4	0.0268	8.949	109.4	5	0.75	1.47	-77.90
	5	0.0222	7.331	555.4	5	0.0229	7.585	164.9	6	3.15	3.46	-70.31

* The UAVs used within each ALNS instance are the average value of 10 runs. Gap (%) $= \frac{\text{ALNS}-\text{MILP}}{\text{MILP}} \times 100\%$.

5.3. Performance of ALNS on Large-Scale Instances

Additionally, the ALNS algorithm is used to solve large-scale instances. Table 5 indicates the results of *Set B* instances with 100 customers. Trips is the number of trips that drones would execute. The ALNS algorithm can find feasible solutions for large instances within a limited time. However, due to its high spatial complexity, the MILP model can't be solved. As shown in Table 5, the average computational time is 689.6 s, indicating the high computational efficiency of the proposed heuristic algorithm.

Table 5. ALNS results for Set B instances with size 100.

Instance	Energy (Wh)	Distance (km)	CPU (s)	Trips *	UAVs *
c201	8.035	2.639	500.0	30	12.3
c202	8.020	2.639	633.6	37.2	14.2
c203	8.002	2.641	754.2	37	16.8
c204	8.007	2.644	870.5	37.1	21.1
Average	8.016	2.641	689.6	37.8	16.1

* The trips and UAVs used within each ALNS instance are the average value of 10 runs.

5.4. Comparison between Variable-Speed and Fixed-Speed

This section provides the energy saving achievable by enabling drones to fly at variable speeds. The MILP implementation is run with varying speeds and three fixed speeds (10/20/30 m/s) to generate solutions for the DRP-VFS. Table 6 shows that the average total energy consumption can be reduced by up to 46.61 % against the 10 m/s case, 1.03 % against the 20 m/s case and 15.13 % against the 30 m/s case. The drone speeds in variable speed cases are all located at maximum-range speed since this is the most energy-efficient speed for drones. The results in Table 6 demonstrate that variable flight speeds result in lower energy consumption than fixed speeds in most instances. Reducing energy

 doi:10.20944/preprints202403.1028.v1

consumption could enable the utilization of smaller-capacity batteries, resulting in weight savings and possibly extending the flight range.

Table 6. Comparison of energy consumption for the variable-speed case versus the fixed-speed cases (10, 20 and 30 m/s).

Customer	Instance	Energy consumption (kWh)			
		10	20	30	variable-speed (% desc. over 10/20/30 m/s)
20	1	0.0571	0.0309	0.0361	0.0306 (-46.39/-0.94/-15.29)
	2	0.0495	0.0265	0.0308	0.0262 (-47.14/-1.15/-14.99)
	3	0.0521	0.0283	0.0331	0.0281 (-46.09/-0.82/-15.23)
	4	0.0496	0.0269	0.0314	0.0266 (-46.34/-1.18/-15.22)
	5	0.0419	0.0224	0.0261	0.0222 (-47.06/-1.06/-14.93)

6. Conclusion and Future Work

This paper extends the multi-trip DRP with time windows to incorporate variable flight speeds. We proposed a MILP formulation for DRP-VFS that minimizes total cost while considering the energy consumption model, flight speed, payload weight, and drone reuse. To tackle practical instances, we proposed a three-phased ALNS heuristic algorithm. Numerical results indicate that optimizing drone speeds is an important consideration for drone delivery. Variable-speed resulted in about 46 % and 15 % improvements in total energy consumption compared to fixed-speed 10 m/s and 30 m/s cases, respectively. Numerical experiments also revealed that the three-phased heuristic performed well in both solution quality and CPU times in small and large DRP-VFS instances.

There are multiple opportunities for further research in this scope. For instance, exact solution methods can be explored to address the nonlinear energy function. To extend the flight range of drones, multiple depots or recharging stations might be investigated. More destruction and repair operations can be applied to improve solution quality. Additionally, the problem could be raised by considering heterogeneous drones or collaborative delivery of drones and trucks.

Author Contributions: Conceptualization, Shaofeng Lu; Investigation, Minling Feng; Methodology, Kunpeng Wu; Project administration, Shaofeng Lu and Zenghao Lu; Resources, Zenghao Lu; Software, Kunpeng Wu, Shaofeng Lu and Minling Feng; Supervision, Shaofeng Lu; Writing – original draft, Kunpeng Wu; Writing – review & editing, Shaofeng Lu.

Funding: This research was funded by the National Science Foundation of Guangdong Province, China under Grant 2023A1515012949; in part by the 2022 Fundamental and Applied Fundamental Research Project of Guangzhou Municipal Basic Research Program under Grant 202201010716

Acknowledgments: In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

References

1. Zipline. Drones to the rescue: Zipline's blood deliveries save lives in Rwanda. http://ruralreporters.com/drones-to-the-rescue-ziplines-blood-deliveries-save-lives-in-rwanda/, 2018.
2. Wing. Google's Wing Gets First-Ever FAA Approval For Commercial Delivery Via Drone. https://interdrone.com/news/googles-wing-gets-first-ever-faa-approval-for-commercial-delivery-via-drone, 2019.
3. Amazon. Amazon receives U.S. regulatory approval to start drone delivery trials. https://www.reuters.com/article/us-amazon-prime-air-idUSKBN25R2NG, 2019.

4. Viloria, D.R.; Solano-Charris, E.L.; Muñoz-Villamizar, A.; Montoya-Torres, J.R. Unmanned aerial vehicles/drones in vehicle routing problems: a literature review. *Int. Trans. Oper. Res.* **2020**, *28*, 1626–1657.
5. Moshref-Javadi, M.; Winkenbach, M. Applications and Research avenues for drone-based models in logistics: A classification and review. *Expert Systems with Applications* **2021**, *177*, 114854. https://doi.org/10.1016/j.eswa.2021.114854.
6. Dorling, K.; Heinrichs, J.; Messier, G.G.; Magierowski, S. Vehicle Routing Problems for Drone Delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2017**, *47*, 70–85. doi:10.1109/tsmc.2016.2582745.
7. Kang, M.; Lee, C. An Exact Algorithm for Heterogeneous Drone-Truck Routing Problem. *Transp. Sci.* **2021**, *55*, 1088–1112.
8. Wang, Z.; Sheu, J.B. Vehicle routing problem with drones. *Transportation Research Part B: Methodological* **2019**, *122*, 350–364. doi:https://doi.org/10.1016/j.trb.2019.03.005.
9. Ermagun, A.; Tajik, N. Multiple-Drones-Multiple-Trucks Routing Problem for Disruption Assessment. *Transportation Research Record* **2022**, *2677*, 725 – 740.
10. Agatz, N.; Bouman, P.; Schmidt, M. Optimization Approaches for the Traveling Salesman Problem with Drone. *Transportation Science* **2018**, *52*, 965–981. doi:10.1287/trsc.2017.0791.
11. Liu, Z.; Sengupta, R.; Kurzhanskiy, A. A power consumption model for multi-rotor small unmanned aircraft systems. 2017 International Conference on Unmanned Aircraft Systems (ICUAS), 2017, pp. 310–315. doi:10.1109/ICUAS.2017.7991310.
12. Zeng, Y.; Xu, J.; Zhang, R. Energy Minimization for Wireless Communication With Rotary-Wing UAV. *IEEE Transactions on Wireless Communications* **2019**, *18*, 2329–2345. doi:10.1109/TWC.2019.2902559.
13. Gao, N.; Zeng, Y.; Wang, J.; Wu, D.; Zhang, C.; Song, Q.; Qian, J.; Jin, S. Energy model for UAV communications: Experimental validation and model generalization. *China Communications* **2021**, *18*, 253–264. doi:10.23919/JCC.2021.07.020.
14. Cheng, C.; Adulyasak, Y.; Rousseau, L.m. Drone routing with energy function : Formulation and exact algorithm. *Transportation Research Part B* **2020**, *139*, 364–387. doi:10.1016/j.trb.2020.06.011.
15. Tamke, F.; Buscher, U. The vehicle routing problem with drones and drone speed selection. *Computers & Operations Research* **2023**, *152*, 106112. doi:https://doi.org/10.1016/j.cor.2022.106112.
16. Otto, A.; Agatz, N.; Campbell, J.; Golden, B.; Pesch, E. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks* **2018**, *72*, 411–458. https://doi.org/10.1002/net.21818.
17. Küçükoglu, I.; Dewil, R.; Cattrysse, D. The electric vehicle routing problem and its variations: A literature review. *Comput. Ind. Eng.* **2021**, *161*, 107650.
18. Moghdani, R.; Salimifard, K.; Demir, E.; Benyettou, A. The green vehicle routing problem: A systematic literature review. *Journal of Cleaner Production* **2021**.
19. Murray, C.C.; Chu, A.G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies* **2015**, *54*, 86–109. https://doi.org/10.1016/j.trc.2015.03.005.
20. Luo, Z.; Liu, Z.; Shi, J. A Two-Echelon Cooperated Routing Problem for a Ground Vehicle and Its Carried Unmanned Aerial Vehicle. *Sensors* **2017**, *17*. doi:10.3390/s17051144.
21. Wang, X.; Poikonen, S.; Golden, B. The vehicle routing problem with drones: several worst-case results. *Optimization Letters* **2017**, *11*, 679–697. doi:10.1007/s11590-016-1035-3.
22. Ha, Q.M.; Deville, Y.; Pham, Q.D.; Hà, M.H. On the min-cost Traveling Salesman Problem with Drone. *Transportation Research Part C: Emerging Technologies* **2018**, *86*, 597–621. doi:https://doi.org/10.1016/j.trc.2017.11.015.
23. Boysen, N.; Briskorn, D.; Fedtke, S.; Schwerdfeger, S. Drone delivery from trucks: Drone scheduling for given truck routes. *Networks* **2018**, *72*, 506–527, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/net.21847]. doi:https://doi.org/10.1002/net.21847.
24. Poikonen, S.; Golden, B.; Wasil, E.A. A Branch-and-Bound Approach to the Traveling Salesman Problem with a Drone. *INFORMS J. on Computing* **2019**, *31*, 335–346. doi:10.1287/ijoc.2018.0826.
25. Deng, X.; Guan, M.; Ma, Y.; Yang, X.; Xiang, T. Vehicle-Assisted UAV Delivery Scheme Considering Energy Consumption for Instant Delivery. *Sensors* **2022**, *22*. doi:10.3390/s22052045.
26. D'Andrea, R. Guest Editorial Can Drones Deliver? *IEEE Transactions on Automation Science and Engineering* **2014**, *11*, 647–648. doi:10.1109/TASE.2014.2326952.

27. Torabbeigi, M.; Lim, G.J.; Kim, S.J. Drone Delivery Scheduling Optimization Considering Payload-induced Battery Consumption Rates. *Journal of Intelligent & Robotic Systems* **2020**, *97*, 471–487. doi:10.1007/s10846-019-01034-w.
28. Poikonen, S.; Golden, B. Multi-visit drone routing problem. *Computers and Operations Research* **2020**, *113*, 104802. doi:10.1016/j.cor.2019.104802.
29. Stolaroff, J.K.; Samaras, C.; Mitchell, A.S.; Ceperley, D.; Neill, E.R.O.; Lubers, A. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. *Nature Communications* **2018**, pp. 1–13. doi:10.1038/s41467-017-02411-5.
30. Jeong, H.Y.; Song, B.D.; Lee, S. Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones. *International Journal of Production Economics* **2019**, *214*, 220–233. https://doi.org/10.1016/j.ijpe.2019.01.010.
31. Choi, Y.; Schonfeld, P.M. Optimization of Multi-package Drone Deliveries Considering Battery Capacity. *Transportation Research Board 96th Annual Meeting* **2017**, p. 16.
32. Murray, C.C.; Raj, R. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies* **2020**, *110*, 368–398. https://doi.org/10.1016/j.trc.2019.11.003.
33. Nishira, M.; Ito, S.; Nishikawa, H.; Kong, X.; Tomiyama, H. An ILP-based Approach to Delivery Drone Routing under Load-dependent Flight Speed. 2022 International Conference on Electronics, Information, and Communication (ICEIC), 2022, pp. 1–4. doi:10.1109/ICEIC54506.2022.9748803.
34. Dukkanci, O.; Kara, B.Y.; Bektaş, T. Minimizing energy and cost in range-limited drone deliveries with speed optimization. *Transportation Research Part C: Emerging Technologies* **2021**, *125*, 1–33. doi:10.1016/j.trc.2021.102985.
35. Raj, R.; Murray, C. The multiple flying sidekicks traveling salesman problem with variable drone speeds. *Transportation Research Part C: Emerging Technologies* **2020**, *120*, 102813. doi:10.1016/j.trc.2020.102813.
36. Beale, E.; Tomlin, J. Special facilities in a general mathematical programming system for nonconvex problems using ordered sets of variables. *Operational Research* **1969**, *69*, 447–454.
37. D'Ambrosio, C.; Lodi, A.; Martello, S. Piecewise linear approximation of functions of two variables in MILP models. *Operations Research Letters* **2010**, *38*, 39–46. doi:https://doi.org/10.1016/j.orl.2009.09.005.
38. Pisinger, D.; Ropke, S. A general heuristic for vehicle routing problems. *Computers & Operations Research* **2007**, *34*, 2403–2435. doi:https://doi.org/10.1016/j.cor.2005.09.012.
39. Solomon, M.M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Oper. Res.* **1987**, *35*, 254–265.